IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Abraham R. MATTHEWS et al.	**Examiner:** BRUCKHART, Benjamin
App. No.: 09/663,483	Group Art Unit: 2146
Filed: September 13, 2000	Conf. No.: 2761
For: SWITCH MANAGEMENT SYSTEM AND METHOD	Docket No.: FORT-000600

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF
IN SUPPORT OF APPELLANT'S APPEAL
TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Sir:

Applicant (hereafter "**Appellant**") hereby submits this Brief in support of its appeal from a decision by the Examiner, mailed October 18, 2007, in the above-captioned application. Appellant respectfully requests consideration of this Appeal by the Board of Patent Appeals and Interferences (the "**Board**") for allowance of the above-captioned patent application.

The claims of the above-captioned patent application were finally rejected by the Examiner in a final Office Action mailed October 18, 2007 (the "**Final Office Action**"). On March 18, 2008, the Appellant submitted a Notice of Appeal (via EFS Web) in the above-captioned patent application concurrently with a Response Under 37 C.F.R. §1.116. Therefore, this is a proper Appeal and Appellant's Brief in support of this Appeal follows.

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest in this Appeal is Fortinet, Inc., the assignee of record of the above-referenced patent application.

RELATED APPEALS AND INTERFERENCES

There are no known appeals or interferences related to this Appeal.

STATUS OF CLAIMS

Claims 1-2, 4-8 and 21-34 are currently pending in the above-captioned patent application. In the Final Office Action, the Examiner (i) rejected claims 1, 2, 4-8, 25-29 and 34 under 35 U.S.C. §102(e) as allegedly being anticipated by US Patent No. 6,674,756 of Rao et al. (hereafter "**Rao**"); and (ii) rejected claims 21-24 and 30-33 as allegedly being unpatentable over Rao in view of US Patent No. 7,096,495 of Warrier et al. (hereafter "**Warrier**").

Claims 1-2 and 4-8 as set forth in the Amendment and Response to Office Action submitted November 6, 2007 (the "**Amendment and Response**"), are the subject of this Appeal. The Claims Appendix below sets forth a copy of the appealed claims.

STATUS OF AMENDMENTS

No amendment was filed subsequent to the Final Office Action. Consequently, claims 1-2 and 4-8 are appealed in the form presented in the Amendment and Response.

SUMMARY OF CLAIMED SUBJECT MATTER

The only independent claim in the above-captioned patent application that is the subject of this Appeal, i.e., claim 1, generally relates to a method of managing a switch to create discrete customized services for customers of a service provider[1]. A switch is provided, which has multiple processor elements (PEs)[2]. Each of the PEs execute a network operating system (NOS), which facilitates creation of discrete customized services for customers of a service provider operating the switch. The NOS creates the discrete customized services for each of the customers by providing each customer with a customized configuration of service object groups[3]. As should be apparent from the usage and context within the above-captioned patent application, "objects" are not things that have physical existence as erroneously assumed by the Examiner, but rather are runtime instances of classes that consists of data (e.g., instance variables) and operations / functionality (e.g., instance methods) associated with the corresponding data[4]. A system virtual router is created on a first PE of the multiple PEs[5]. Creation of the system virtual router includes establishing a global object manager associated with the NOS of the first PE[6]. The global object manager is responsible for managing global

---

[1] A switch, e.g., switch 12, of an IP Service Delivery Platform 10 includes a network operating system, e.g., NOS 20, which dynamically distributes different configurations of service object groups to processors of the switch to provide customers with discrete customized services (see, e.g., Specification, ¶ [0047]; Specification, ¶ [0053]; FIG. 1 and FIG. 3). Examples of customized services include managed, network security services and computationaly-intense IP services, such as route forwarding, encryption, managed firewall services and Virtual Private Networks (VPNs) (see, e.g., Specification, ¶ [0040], Specification, ¶ [0043] and Specification, ¶ [0045]).

[2] See, e.g., Abstract.

[3] See, e.g., Specification, ¶ [0046] (switch 12 has the "ability to turn IP services into discrete and customized objects"); Specification, ¶ [0053] ("network operating system 20 enables switch 12 to create discrete customized services to specific subscriber corporations by providing them each with a different configuration of service object groups"); Specification, ¶ [0059] ("[o]bjects represent a basic unit of management"); and FIG. 5.

[4] See, e.g., Specification, ¶ [0065] ("A Group encompasses objects, which are located in different address spaces.") and Specification, ¶ [0066] ("There can be multiple objects of the same class in an object group").

[5] See, e.g., Abstract and Specification, ¶ [0008].

[6] Various components of a network operating system (NOS) 20, including an object manager 24, are shown in FIG. 3. In one embodiment, the object manager 24 includes the three layers shown in FIG. 4, i.e., an OM Controller and

object groups and global object configurations[7]. The multiple PEs are configured from the system virtual router[8]. Configuration of the multiple PEs involves establishing, via the global object manager, a local object manager on each of the multiple PEs. The local object manager of each PE manages objects local to the PE and transfers messages between objects on the PE and between objects on the PE and objects on other PEs[9].

As discussed in Specification, ¶ [0040], service providers need a service delivery mechanism that minimizes capital and operational costs while enabling high-margin, value-added public network services that are easily provisioned, managed and repeated. The IP Service Delivery Platform 10 and the switches 12 depicted in FIG. 1 are part of one solution. As discussed in Specification, ¶ [0204], an IP Service Delivery Platform 10, including an IP Service Processing Switch, such as switch 12, operating as claimed, enables service providers to leverage the enterprise's existing services infrastructure (e.g., leased lines and Frame Relay PVCs) to deliver new, value-added services without the requirement of a truck roll. Additionally, all security service functionality, e.g., firewall and VPN functionality, resides on the IP Service Processing Switch 12 at the POP, thus freeing the service provider from onsite

---

Database (OMCD) 40, an OM Object Routing and Interface Global (OMORIG) 42 and an OM Object Routing and Interface (OMORI) 44.

[7] In one embodiment, the middle layer of the object manager 24, the OM Object Routing and Interface Global (OMORIG) 42 is said to be "concerned with managing global (across the switch system) object groups and objects configurations" (see FIG. 4 and Specification, ¶ [0057]). See also, Specification, ¶ [0058] ("the IPSX object database consists of two types of databases: Global (managed on Master Control Blade by OMORIG) and distributed local databases (managed by OMORI agents on every PE present in the system).").

[8] According to the Specification, the switch is said to have a plurality of processor elements (PEs), which are configured from the system virtual router (see, e.g., Specification, ¶ [0008]).

[9] In one embodiment, the lower layer of the object manager 24, i.e., the OM Object Routing and Interface (OMORI) 44 is said to be "concerned with managing local objects and groups as well as routing control information between address spaces based on location of objects, and interfacing with the object via method invocation" (see FIG. 4 and Specification, ¶ [0057]). See also, Specification, ¶ [0058] ("the IPSX object database consists of two types of databases: Global (managed on Master Control Blade by OMORIG) and distributed local databases (managed by OMORI agents on every PE present in the system).") and Specification, ¶ [0078] ("OMORI is the OM agent. OMORI runs on every processing node and manages local objects and forwards IOCTLs to another object, whether local or remote.").

systems integration and configuration and effectively hiding the technology from the enterprise customer.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Did the Examiner improperly reject claims 1-2 and 4-8 under 35 U.S.C. §102(e) by attributing capabilities and functionality to Rao that are clearly unsupported by and outside of the scope and contemplation of Rao?

ARGUMENT

A. The Examiner improperly rejected claims 1-2 and 4-8 under 35 U.S.C. §102(e) by attributing to Rao capabilities and functionality that are neither required, taught, nor reasonably suggested by the disclosure of Rao.

Claims 1-2 and 4-8

In the Final Office Action, the Examiner incorrectly rejected claims 1-2 and 4-8 under 35 U.S.C. §102(e) as being anticipated by Rao. It is respectfully submitted that the Examiner has failed to establish a *prima facie* case of anticipation. The Examiner has the initial burden of establishing a *prima facie* case of anticipation by pointing out where all of the claim limitations appear in a single reference. See In re Spada, 911 F.2d 705, 709, 15 USPQ2d 1655 (Fed. Cir. 1990). In re King, 801 F.2d 1324, 231 USPQ 136 (Fed. Cir. 1986). Furthermore, a rejection for anticipation under §102 requires that the four corners of a single prior art document describe every element of the claimed invention, either expressly or inherently, such that a person of ordinary skill in the art could practice the invention without undue experimentation. In re Paulsen, 30 F.3d 1475, 1478-79, 31 USPQ2d 1671, 1673 (Fed. Cir. 1994).

In the case of claim 1, the Examiner's broad brush stroke approach leaves many gaps. The Examiner has pointed to no reasonable support in Rao for any of the following elements of claim 1: (i) a network operating system (NOS) on each of a plurality of PEs of a switch supporting the creation of discrete customized services for customers of a service provider operating the switch by providing each customer with a customized configuration of service object groups; (ii) a global object manager as recited; and (iii) local object managers established on each of the PEs as recited.

For example, in the Final Office Action, the Examiner cites to col. 4, ll. 1-5 and col. 8, l. 38 to col. 9, l. 43 of Rao to support his position that Rao teaches a network operating system (NOS), running on a plurality of processor elements (PEs) of a switch, to allow the switch to "create discrete customized services for customers … ***by providing each customer with a customized configuration of service object groups***" (emphasis added). The portions of Rao relied upon by the Examiner are reproduced below for the Board's convenience.

Rao, col. 4, ll. 1-5:

> interface module (a card), referred to as a forwarding module (FM) 10. Each FM 10 preferably includes the on-board intelligence, route forwarding,, and route processing information for distributed packet forwarding, as is described in further detail below.

Rao, col. 8, l. 38 to col. 9, l. 43:

> Referring again to FIG. 2, each FM 10 further includes a connection manager 46 and a resource manager 38. The connection manager 46 detects incoming calls to the FM 10 and the resource manager 38 manages and allocates local resources including digital modem and ISDN switched resources. Each connection to the switch needs a specific set of hardware and software resources. A frame relay call, for example, needs a line interface, an HDLC controller, a frame relay protocol stack, and frame forwarding software. Generally, all the resources required for a connection are found on the input FM 10 and its associated PMs 12. Sometimes, however, traffic entering the system on one card requires resources on another. Thus, when the connection manager 46 detects an incoming call, a resource request is broadcast over the cell bus 20. The resource manager 38 in each card receives the request and determines what resources are needed. If the card has the requested resource, it is allocated to the incoming call.
>
> Each FM 10 also includes an IP forwarder 44 for forwarding packets based on layer three addresses. The IP forwarder module preferably contains local routing information for forwarding a packet via a right, or first, IP forwarding engine 42a and a left, or second, IP forwarding engine 42b. When a packet is received by the FM 10, the IP forwarder 44 proceeds to forward the packet if it has learned the destination address. Otherwise, the IP forwarder 44 performs a lookup of a central routing table and obtains the necessary routing information.
>
> FIG. 3 is an exemplary flow diagram for processing a connection request coming into the switch of FIG. 1. The program starts, and in step 50, the connection

> manager 46 detects an incoming call in one of the physical ports of the FM 10 (the receiving FM). In step 52, the connection manager 46 notifies the resource manager 38 in the receiving FM 10 of the incoming call. The resource manager 38, in step 54, searches a call policy database for a call policy record corresponding to the incoming call. The call policy record includes various parameters which dictate how the call is to be routed. Different policies may be applied based on the inlink of the call, a telephone number, a domain name, a source address, a destination address, and the like.
>
> Included in the call policy parameters are a quality of access (QoA) level, quality of service (QoS) level, virtual router ID, and virtual private network ID associated with the call. QoA is a method of classifying users and granting access to the switch based on a comparison of their QoA level to the current resource utilization. This allows for tiered access to the Internet when there s a competition for resources. Each QoA level is preferably assigned a percentage of threshold resource usage. If resource utilization is below the percentage of threshold resource usage assigned to the incoming call's QoA level, the call is accepted. Otherwise, the call is rejected.
>
> QoS is a method of classifying users to determine the priority with which packets are conveyed once a call has been accepted. QoS offers preferential treatment by processing connections based on their QoS levels. The higher the QoS level attached to the call, the higher the processing priority given to the packets associated with the call.
>
> The incoming call's virtual router ID and virtual private network ID allow the switch to provide access to resources that the user authorized for. According to one embodiment of the invention, the switch may be partitioned into multiple virtual routers where each virtual router has its own set of resources (e.g. ISDN or modem resources) and routing tables. Thus, each virtual router preferably functions as a separate router in an independent and self-contained manner. Each virtual router may further be partitioned into multiple virtual private networks (VPNs) for further controlling access to the switch. VPNs are created with filtering software that filters traffic directed to the virtual router based on criteria such as, for example, source address and/or destination address.

As the undersigned attempted to explain in the Response After Final, these portions of Rao merely include a discussion relating to the connection manager 46, resource manager 38 and IP forwarder 44 of Fig. 2 and the flow diagram of Fig. 3. The undersigned can find no reasonable relationship between the relied upon portions of Rao and the recited network operating system of claim 1, which allows the switch to "create discrete customized services for

customers … by providing each customer with a ***customized configuration of service object groups***" (emphasis added). As best understood by the undersigned, the Examiner appears to erroneously equate the recited "customized configuration of service object groups" with the physical resources, such as, digital modems and ISDN switched resources, which are allocated to incoming calls in the context of Rao.

As the undersigned indicated in the Response After Final, the relied upon portions of Rao make no mention of a network operating system and make no mention or suggestion regarding the apparently inherent network operating system providing customers with a customized configuration of service object groups. The only use of the term "object" within the entire disclosure of Rao is with respect to the switch creating "a port interface (PIF) 122 object" (see, e.g., col. 10, ll. 50-51). As best as can be understood by the undersigned, the Examiner's reasoning in the Final Office Action with respect to the first element of claim 1 amounts essentially to: (i) an operating systems is an "inherent feature on computers;" and (ii) this inherent feature of Rao allows the switch of Rao to create discrete customized services for customers by providing each customer with a customized configuration of unmentioned (and presumably again, inherent) service object groups. Since the Examiner has failed to point out where in Rao it is taught to provide each customer with a customized configuration of service object groups as required by claim 1, for at least this reason, he has failed to make out a *prima facie* case of anticipation.

In the Final Office Action, other than simply citing to a portion of Rao mentioning "a default system router," the Examiner also failed to point out where in Rao a global object manager, which is "responsible for managing global object groups and global object configurations" is taught or reasonably suggested. The Examiner relies upon col. 19, ll. 39-43 of

Rao for the alleged teachings regarding the recited "global object manager," but does not explain how he apparently equates the default system router of Rao to the global object manager required by claim 1. As indicated above and for the Board's convenience, the undersigned notes that an example of the recited global object manager is the OM Object Routing and Interface Global component 42 of the network operating system shown in FIG. 4.

In the Final Office Action, the Examiner additionally failed to point out where in Rao the local object managers, which manage objects local to the given PE and transfer messages between objects on the given PE and between objects on the given PE and objects on other PEs of the plurality of PEs, are taught or reasonably suggested. The Final Office Action includes a citation to Rao, col. 8, ll. 38-55, but no correspondence between the resource manager 38 and its allocation of resources to calls and the functionality of the local object manager at issue is otherwise drawn. In claim 1, the local object manager both "manages ***objects*** local to the given PE" and "***transfers messages between objects***" (emphasis added) on the same PE and objects on different PEs. For the Board's convenience, the undersigned notes that an example of the claimed functionality is illustrated in FIG. 4 in which OM Object Routing and Interface components 44 on different PEs are shown transferring messages among objects of an object group. For the Board's further convenience, the undersigned again points out "objects" in the context of the above-captioned patent application are "objects" are runtime instances of classes – not things having physical existence as presently assumed by the Examiner.

In summary, while ***Rao*** is understood to teach a resource manager 38 on each card that ***can allocate physical resources*** (such as digital modems and ISDN switched resources) ***to incoming calls***, Rao clearly lacks teaching of the particular mechanisms and steps through which discrete customized services are provided to each customer of a service provider operating a

switch performing the recited method of claim 1. The undersigned cannot understand how allocation of physical resources to calls relates to the claimed method, which involves (i) a network operating system supporting the creation of discrete customized services for multiple customers of a service provider by providing each customer with a customized configuration of service object groups and (ii) the distribution and management of the objects by a global object manager and a local object manager.

As evidenced by the foregoing, the Examiner has incorrectly attributed teachings to Rao that are clearly absent from and not contemplated by the disclosure of Rao. The Examiner then proceeds to use such attributed teachings to improperly find anticipation under 35 U.S.C. §102(e). For at least these reasons, the undersigned respectfully requests the Board to reverse the Examiner's rejections of claims 1-2 and 4-8.

CONCLUSION

The Examiner has failed to establish a *prima facie* case to support his 35 U.S.C. §102(e) rejections. Rao does not teach or reasonably suggest at least (i) a network operating system (NOS) on each of a plurality of PEs of a switch supporting the creation of discrete customized services for customers of a service provider operating the switch by providing each customer with a customized configuration of service object groups; (ii) a global object manager as recited by claim 1; and (iii) local object managers established on each of the PEs as recited by claim 1. The Examiner has improperly attributed teachings and/or functionality to Rao that are unsupported by, inconsistent with, not enabled by and outside the scope of the written description of Rao. For the aforementioned reasons, the Examiner's rejections should be reversed, and claims 1-2 and 4-8 should be allowed.

Respectfully submitted,

HAMILTON, DESANCTIS & CHA

Date July 30, 2008

By: /Michael A. DeSanctis/

Michael A. DeSanctis, Esq.
Reg. No. 39,957
Customer No. 64128
Ph: (303) 856-7155

CLAIMS APPENDIX

1. A method comprising:

providing a switch having a plurality of processor elements (PEs), each of the plurality of PEs running a network operating system (NOS), the NOS allowing the switch to create discrete customized services for customers of a service provider operating the switch by providing each customer with a customized configuration of service object groups;

creating a system virtual router on a first PE of the plurality of PEs, wherein creating the system virtual router includes establishing a global object manager associated with the NOS of the first PE, the global object manager being responsible for managing global object groups and global object configurations; and

configuring the plurality of PEs from the system virtual router, wherein configuring includes establishing, via the global object manager, a local object manager on each of the PEs, wherein the local object manager for a given PE of the plurality of PEs manages objects local to the given PE and transfers messages between objects on the given PE and between objects on the given PE and objects on other PEs of the plurality of PEs.

2. An article comprising a computer readable medium having instructions thereon, wherein the instructions, when executed in a computer, create a system for executing the method of claim 1.

4. The method of claim 1, wherein said configuring PEs of the plurality of PEs includes creating a customer virtual router from selected PEs on multiple blades of the switch, wherein creating a customer virtual router includes:

establishing a virtual private network (VPN) associated with a customer;

adding the customer virtual router to a list of virtual routers associated with the VPN; and

creating an object associated with the customer virtual router on each of the selected PEs.

5. The method of claim 1, wherein said configuring the plurality of PEs includes:

adding new PEs; and

using a distributed management layer to group PEs into at least one virtual router, wherein grouping includes assigning a group identifier to selected objects in each PE such that the selected objects can be addressed as a group.

6. The method of claim 5, wherein using a distributed management layer to group processor elements into at least one virtual router includes:

requesting the global object manager to create a virtual router from a group of PEs;

requesting one or more of the local object managers to group the group of PEs;

activating PEs of the group; and

generating a status message that the at least one virtual router is created.

7. The method of claim 6, wherein said activating PEs of the group includes causing a state machine for a PE of the PEs of the group to enter an active state.

8. The method of claim 5, wherein said using a distributed management layer to group PEs includes adding object identifiers to a global object database.

EVIDENCE APPENDIX

NONE.

<u>RELATED PROCEEDINGS APPENDIX</u>

NONE.